WE CLAIM:

1. A data processing apparatus, comprising:

a data processing unit operable to perform data processing operations on data values;

a register file having a plurality of registers operable to store said data values for access by the data processing unit;

the data processing unit being responsive to a single transfer instruction to perform multiple data value transfers between a corresponding multiple of said registers of said register file and consecutive data value addresses in a memory, the single transfer instruction providing an address identifier from which said consecutive data value addresses are derivable, and further providing for each of said data value transfers a register identifier identifying the register within said plurality of registers which is the subject of that data value transfer, said register identifier for each of said data value transfers being specifiable independently of the register identifiers specified for the other of said data value transfers.

2. A data processing apparatus as claimed in Claim 1, wherein said single transfer instruction is a load instruction, the data processing unit being responsive to the load instruction to perform said multiple data value transfers from the consecutive data value addresses in said memory to said corresponding multiple of said registers of said register file.

3. A data processing apparatus as claimed in Claim 1, wherein said single transfer instruction is a store instruction, the data processing unit being responsive to the store instruction to perform said multiple data value transfers from said corresponding multiple of said registers of said register file to the consecutive data value addresses in said memory.

4. A data processing apparatus as claimed in Claim 1, wherein the address identifier comprises a base address and an offset value.

5. A data processing apparatus as claimed in Claim 4, wherein the base address is specified within the single transfer instruction by a base address register identifier identifying one of said plurality of registers that is arranged to store the base address.

6. A data processing apparatus as claimed in Claim 4, wherein the offset value is specified within the single transfer instruction by an offset register identifier identifying one of said plurality of registers that is arranged to store the offset value.

7. A data processing apparatus as claimed in Claim 4, wherein the offset value is specified by an immediate value provided within the single transfer instruction.

8. A data processing apparatus as claimed in Claim 1, wherein the data processing unit is responsive to the single transfer instruction to perform two data value transfers.

9. A data processing apparatus as claimed in Claim 1, wherein each of said data values comprise a 32-bit data word, and said consecutive data value addresses identify addresses for a series of adjacent 32-bit data words in the memory.

10. A data processing apparatus as claimed in Claim 1, further comprising an interface between said register file and said memory which facilitates the performance of said multiple data value transfers in parallel.

11. A method of operating a data processing apparatus to transfer data values between a register file and a memory, the register file having a plurality of registers operable to store said data values for access by a data processing unit operable to perform data processing operations on said data values, the method comprising the steps of:

in response to a single transfer instruction, performing multiple data value transfers between a corresponding multiple of said registers of said register file and consecutive data value addresses in a memory by:

deriving said consecutive data value addresses from an address identifier provided by the single transfer instruction;

determining for each of said data value transfers, with reference to a corresponding register identifier provided by said single transfer instruction, the register within said plurality of registers which is the subject of that data value transfer, the register identifier for each of said data value transfers being specifiable independently of the register identifiers specified for the other of said data value transfers; and

performing the multiple data value transfers.

12. A method as claimed in Claim 11, wherein said single transfer instruction is a load instruction, in response to the load instruction, the method performing said multiple data value transfers from the consecutive data value addresses in said memory to said corresponding multiple of said registers of said register file.

13. A method as claimed in Claim 11, wherein said single transfer instruction is a store instruction, in response to the store instruction the method performing said multiple data value transfers from said corresponding multiple of said registers of said register file to the consecutive data value addresses in said memory.

14. A method as claimed in Claim 11, wherein the address identifier comprises a base address and an offset value.

15. A method as claimed in Claim 14, wherein the base address is specified within the single transfer instruction by a base address register identifier identifying one of said plurality of registers that is arranged to store the base address.

16. A method as claimed in Claim 14, wherein the offset value is specified within the single transfer instruction by an offset register identifier identifying one of said plurality of registers that is arranged to store the offset value.

17. A method as claimed in Claim 14, wherein the offset value is specified by an immediate value provided within the single transfer instruction.

18. A method as claimed in Claim 11, wherein in response to the single transfer instruction, the method performs two data value transfers.

19. A method as claimed in any Claim 11, wherein each of said data values comprise a 32-bit data word, and said consecutive data value addresses identify addresses for a series of adjacent 32-bit data words in the memory.

20. A method as claimed in Claim 11, wherein said multiple data value transfers between said register file and said memory are performed in parallel.

21. A computer program product having a computer program executable on a data processing apparatus having a data processing unit operable to perform data processing operations on data values and a register file having a plurality of registers operable to store said data values for access by the data processing unit, the computer program including a single transfer instruction which when executed on the data processing apparatus is operable to cause multiple data value transfers between a corresponding multiple of said registers of said register file and consecutive data value addresses in a memory by:

deriving said consecutive data value addresses from an address identifier provided by the single transfer instruction;

determining for each of said data value transfers, with reference to a corresponding register identifier provided by said single transfer instruction, the register within said plurality of registers which is the subject of that data value transfer, the register identifier for each of said data value transfers being specifiable independently of the register identifiers specified for the other of said data value transfers; and

performing the multiple data value transfers.

22. A computer program product as claimed in Claim 21, wherein said single transfer instruction is a load instruction which when executed on the data processing apparatus is operable to cause said multiple data value transfers to be performed from the consecutive data value addresses in said memory to said corresponding multiple of said registers of said register file.

23. A computer program product as claimed in Claim 21, wherein said single transfer instruction is a store instruction which when executed on the data processing apparatus is operable to cause said multiple data value transfers to be performed from said corresponding multiple of said registers of said register file to the consecutive data value addresses in said memory.

24. A computer program product as claimed in Claim 21, wherein the address identifier comprises a base address and an offset value.

25. A computer program product as claimed in Claim 24, wherein the base address is specified within the single transfer instruction by a base address register identifier identifying one of said plurality of registers that is arranged to store the base address.

26. A computer program product as claimed in Claim 24, wherein the offset value is specified within the single transfer instruction by an offset register identifier identifying one of said plurality of registers that is arranged to store the offset value.

27. A computer program product as claimed in Claim 24, wherein the offset value is specified by an immediate value provided within the single transfer instruction.

28. A computer program product as claimed in Claim 21, wherein when the single transfer instruction is executed on the data processing apparatus, two data value transfers are performed.

29. A computer program product as claimed in Claim 21, wherein each of said data values comprise a 32-bit data word, and said consecutive data value addresses identify addresses for a series of adjacent 32-bit data words in the memory.

30. A computer program product as claimed in Claim 21, wherein said multiple data value transfers between said register file and said memory are performed in parallel.

31. A computer program operable to configure a data processing apparatus to perform a method as claimed in Claim 11.

32. A carrier medium comprising a computer program as claimed in Claim 31.